8-1-2001

2001 Computer Science Department MQP Review

Micha Hofri
Worcester Polytechnic Institute, hofri@cs.wpi.edu

Craig E. Willis
Worcester Polytechnic Institute, cew@cs.wpi.edu

WPI-CS-TR-01-18 August 2001

2001 Computer Science Department MQP Review

by

Micha Hofri
Craig E. Wills

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Abstract

This report presents results of a peer review of MQPs conducted within the Computer Science Department during the Summer of 2001 as part of a campus-wide MQP review. The goal of the report is to assess whether the department MQPs are accomplishing their educational goals. An additional goal of this year's report is to examine the MQP role in fulfilling department-defined student learning outcomes.

The report identifies problems that need to be addressed and trends that need to be continued to make the MQPs a worthwhile learning experience. It reflects data and evaluations for 47 MQPs, involving 104 computer science students, that were completed between the Summer of 2000 and the Spring of 2001. The report also makes comparisons to similar reviews done in the past.

Overall, the large majority of the projects are meeting the educational goals of the department. The reviews indicate that the overall quality of the projects is up a bit from the 1999 MQP Review with 92% of the projects rated as at least adequate with 19% judged to be excellent. The report draws a number of conclusions about the success of the projects based upon the data collected and evaluations done for this review. The report concludes with recommendations for future reviews as the department continues to use the MQP Review as part of a larger department assessment effort.

1 Introduction

1.1 Purpose

The Major Qualifying Project (MQP) is required of all undergraduate students at Worcester Polytechnic Institute. The MQP within the Computer Science Department is a capstone experience, requiring one unit of work, that gives students practice on applying the fundamentals and skills they have learned to a large problem in the field of Computer Science. The project may involve original research, data collection, analysis, or design of a system and often a software implementation. The approach is determined by the student/advisor team. The MQP allows students to study an area of Computer Science in depth, or allows them to combine areas into a single project.

This report presents results of the sixth biennial peer review of MQPs conducted within the Computer Science Department during the Summer of 2001 as part of a campus-wide MQP review. The goal of the report is to assess whether the department MQPs are accomplishing their educational goals. The report identifies problems that need to be addressed and trends that need to be continued to make the MQPs a worthwhile learning experience. It reflects data and evaluations for 47 MQPs, involving 104 computer science students, that were completed between the Summer of 2000 and the Spring of 2001. The report makes comparisons to the following reviews:

Year	Number of MQPs	Number of Students
1991	19	31
1993	26	44
1995	23	43
1997	29	57
1999	31	65
2001	47	104

1.2 MQP Role in Student Outcome Assessment

Examination of student learning outcomes is a point of emphasis for the campus as a whole and it is particularly important to the computer science department as it seeks to measure the effectiveness of its degree program.

The MQP is the capstone component in the degree program so it is natural to use it for measuring student learning outcomes.

During the 2000-01 academic year, the department worked on a draft document containing objectives and outcomes for the Computer Science Undergraduate Program. Although the document has not yet been approved by the department, data collected during this review process is used to determine if relevant draft learning outcomes were exhibited by MQP project members.

1.3 Procedure

The peer review was conducted during the Summer of 2001 by Micha Hofri, department head, and Craig E. Wills, associate professor. The review was to be for projects completed during the 1999-2000 and 2000-01 academic years. Rather than examine a sampling of reports for a two-year period, the peer review team examined projects completed between the Summer of 2000 and the Spring of 2001. This methodology is consistent with past MQP review practice. The report for each MQP was obtained from either the project advisor or from the Gordon Library. Additional project information was gathered from CDR (Completion of Degree Requirement) records.

As a consequence of feedback from the 1999 review process, the procedure for this review process was modified to include faculty advisor input on some aspects of each project. The peer review team used a similar process from previous years [3, 4, 5, 1, 2]. This approach was used to ensure longitudinal analysis of results with previous years, although some changes were made in the questions to aid in the assessment of learning outcomes.

The project review form in Appendix A was separately completed for each project by both a faculty advisor and by one of the reviewers. This process was used to gather feedback from faculty advisors and allow reviewer rankings to be compared with those of the advisor. The form contains information used in classifying the projects, questions quantified on a scale between 1 and 5, and has questions for written comments concerning the report. The form was designed to be easy to fill out with information that could be quickly collected and compared. Written comments concerning the report were used to gather more detailed information about the project and give a means to express specific project strengths and weaknesses. A project evaluation form was received back from a faculty advisor for each project reviewed.

In addition to the first review form, the reviewers filled out an additional one page form shown in Appendix B. This form gathered additional information about the project including an overall assessment of project and report quality. Project grades and registration information was obtained from CDR records. Grades were not consulted until after the MQPs were reviewed.

The MQP reports were divided between the two reviewers for evaluation. After all evaluations were completed by the reviewers and the faculty advisors, the data from the forms were collected and analyzed. This report is the outcome of the peer review process. Section 2 presents the results from the evaluation forms. Section 3 analyzes and correlates the results. Section 4 analyzes the results relative to desired student outcomes for the undergraduate computer science degree program. Section 5 discusses conclusions and recommendations.

2 Results

This section presents the results of the Computer Science MQP evaluations. Along with presentation of the results are included reviewer comments (denoted by **Comment:**) which highlight the results and contrast them against those from previous reviews when appropriate. Note: All data are presented on a per project and not per student basis.

All percentages are represented in whole number amounts (i.e., 1/31 is represented as 3%), and all number averages are represented to one decimal accuracy (i.e., 1.97 is shown as 2.0). Because of this format, the percentages do not always total to 100%.

2.1 Faculty/Student Ratio

Table 1 shows the percentage of projects with the given numbers of students and faculty. None of the projects were completed with students from other departments, nor were any faculty from outside of the department listed as official project advisors.

The average number of students per project was 2.3. The average number of faculty per project was 1.3.

Comment: The results show that thirteen (28%) of the projects were done by a single student. The number of such projects is up, but the percentage of projects is down from the 1999 figure of 35%. The average number of students per project increased slightly to 2.3 from 2.1 in 1999 with the number of three- and four-student team projects increasing from 38% to 51%. The number of projects advised by a single faculty member was 34 (72%), virtually the same as in previous years.

Table 1: Percentage of Projects with the Given Number of Students and Faculty

	Students				
Faculty	**1**	**2**	**3**	**4+**	**Total**
1	21	15	30	6	72
2	6	6	11	0	23
3+	0	0	0	4	4
Total	28	21	40	11	100

2.2 Faculty Project Load

Table 2 shows the distribution on the number of projects (co-)advised by each faculty member. There were 17 full-time faculty in Computer Science during AY00-01 (one faculty was on sabbatical) plus one associated faculty member and one professor of practice who advised projects. Table 3 shows the same data, but with per-advisor weights of 1/2 or 1/3 for projects with two and three advisors. Note: Loads for co-advisors from other departments are not shown in the tables.

Comment: The average project load increased to 2.8, from 2.4 in 1999. The comparable average loads shown in Table 3 increased to 2.3 projects/faculty in 2001 from 1.8 projects/faculty in 1999. These numbers are expected given the significant increase in projects this academic year.

The Gini Coefficient, a number between zero and one was also calculated for Tables 2 and 3. This coefficient measures the degree to which projects are evenly distributed amongst faculty with a coefficient of zero indicating perfect distribution and a value of one indicating all projects being advised by a single faculty member. Table 4 shows comparable figures for all years in which this coefficient has been computed. The results indicate an improvement over

Table 2: Distribution of Projects Advised or Co-advised

Number of Projects (Co-)Advised	Number of Faculty
0	1
1	7
2	1
3	3
4	4
5	0
6	2
7	0
8	1

avg: 2.8 projects/faculty

Table 3: Distribution of Load of Projects Advised

Load of Projects Advised	Number of Faculty
0	1
0.5	3
1	4
2	1
2.5	2
3	2
3.2	1
3.5	3
4.5	1
7	1

avg: 2.3 projects/faculty

previous years in more evenly distributing the projects amongst department faculty. This improvement is more encouraging given the existence of two off-campus project centers where all Computer Science projects at each site were advised by the same faculty member.

Table 4: Gini Coefficient for Project Advising Load Amongst Faculty

Year	Number of Projects (Table 2)	Load of Projects (Table 3)
1997	0.484	0.554
1999	0.406	0.453
2001	0.402	0.399

2.3 Off-Campus Projects

Eleven (23%) of the projects were sponsored or involved off-campus companies and organizations. The sponsors were Goddard Space Flight Center (3), Natural Microsystems, Atmel Corporation (2), General Dynamics, SRI, Sybase, Bose Corporation, and the University of Siegen, Germany. The remaining projects were done on-campus and not sponsored by off-campus companies.

Comment: The number of off-campus sponsored projects increased from 16% in 1999. This number is comparable to a figure of 24% in 1997.

2.4 Project Grades

In the projects reviewed, 66% of the projects (65% of the students) received a final grade of A, 32% of the projects (33% of the students) received a final grade of B, and 3% of the projects (2% of the students) received a final grade of C. Note: two projects resulted in members on a given project receiving different individual grades. For purposes of this review, the project grade used for these projects was the average of the individual student grades.

Comment: The number of A grades given to projects dropped from previous years where over 70% of projects and students received this grade. The number of C grades was comparable to 1999 when 3% of the projects

(6% of the students) earned that grade. Table 5 shows the distribution of project grades found during each MQP review.

Table 5: Percentage of MQP Project Grades for Each MQP Review

Year	Project (Student) Grades		
	A	B	C
1991	58 (71)	42 (29)	0 (0)
1993	69 (73)	23 (20)	8 (7)
1995	63 (60)	22 (30)	15 (9)
1997	72 (71)	27 (21)	14 (7)
1999	71 (77)	26 (17)	3 (6)
2001	66 (65)	32 (33)	3 (2)

2.5 Project Continuation

Thirteen projects (28%) were continuations of prior MQPs and MS/PhD theses. The other projects were not directly related to other projects.

Comment: These numbers are consistent with 1999 when 29% of projects were continuations. In 1995 and 1997 only 14% and 7% of the projects were continuations. The results indicate that faculty have continued to integrate new projects with previous work.

2.6 Project Duration

Table 6 shows the duration of each project. Due to difficulties in obtaining registration data, project duration data is not known for all projects. The data show a variety of combinations for the number of terms and the amount of unit registration. Note the table shows *registered* units and not *earned* units so that a project needing an extra one-sixth unit to complete the project may not correspond to earned credit.

Comment: Past MQP reviews used earned credit rather than registered units so it is difficult to compare these data with previous years.

Table 6: Percentage of Projects with the Given Duration in Terms and Registered Units

	Units			
Terms	**1**	**1 1/6+**	**unk.**	**Total**
1	6	0	0	6
2	15	2	0	17
3	26	6	0	32
4	4	28	0	32
unknown	0	0	13	13
Total	51	36	13	100

2.7 Project Report Size

The average size of the project reports was 58 pages (with a range of 20–115), which excludes appendices and code.

Comment: The length of reports is about the same as previous years: 45 (1991), 49 (1993), 50 (1995), 59 (1997) and 50 (1999).

2.8 References

The average number of references was 17 (with a range of 0–40) for each report. Results reported in Table 12 show 13% of the project reports were less than adequate with 49% better than adequate in terms of the quality of the background and literature review.

Comment: This number is up from an average of 12 in 1999. In 1999, 25% of the project reports exhibited a missing or poor literature review so the 2001 results are an improvement.

2.9 Type of Projects

Table 7 shows the percentage of projects that involved different types of work. In many cases a project involved only one area while in other cases it involved multiple areas (thus the percentages total to over 100%).

Comment: As in previous years a significant number of the projects involved a design component and in most cases implementation of a program. The number of projects involving design and implementation of a piece of software is comparable to previous reviews. The collected data was primarily from advisor feedback so the types of work is more detailed than in past review cycles.

Table 7: Types of Work on Projects by Percentage

87	Design/Implementation
45	Research
43	Analysis
40	Performance Evaluation
32	Data Collection (Empirical)
6	Simulation
4	Design
2	Survey

2.10 Project Area

Table 8 shows the percentage of projects that involved different areas of Computer Science. In some cases a project involved only one area while in other cases it involved multiple areas (thus the percentages total to over 100%).

Comment: As the data show there is a variation in the sub-areas of Computer Science covered by the projects, with human-computer interaction and software engineering being involved in many projects.

2.11 Software Used

Table 9 shows the relative use of different programming languages and other software in the projects. Some projects used more than one software tool thus the percentages total to over 100%.

Comment: The use of the Java programming language is the highest, but dropped from a value of 48% in 1999. C and C++ continue to be used

Table 8: Project Areas by Percentage

34	HCI
34	Software Engineering
19	DataBase
19	Operating Systems
17	Networks
17	Webware
13	Data Mining
13	Languages/Compilers
11	Graphics
9	Distributed Systems
6	Architecture
6	Theory/Foundations
4	AI
11	Other (Vision, Signal Analysis, Security, Image Processing, Embedded Systems)

in many projects. Scripting languages, primarily Perl, are also used in many projects. The use of Visual Basic dropped from 13% in 1999 to 4% in 2001.

2.12 Hardware Used

Table 10 shows the percentage of projects that used different types of hardware platforms for their work. The numbers do not add to 100% since some projects involved multiple hardware platforms.

Comment: The data show that about half of the projects used a PC/Windows environment with about one-fourth of the projects using a PC/Linux platform. Feedback from advisors and a distinction between PC/Windows and PC/Linux, made these data much easier to gather in this year's review.

2.13 Advisor/Reviewer Project Evaluations

Numerical evaluations of the projects are shown in Table 11 based on the questions from the form in Appendix A. This form was completed by both

Table 9: Software Used by Percentage

36	Java
32	C
30	C++
23	Perl
4	Visual Basic
2	Assembly Lang.
2	Lisp/Scheme
4	Other (Rational Rose, Oracle)

Table 10: Hardware Used by Percentage

51	PC/Windows
23	PC/Linux
15	CCC Unix
11	CS Unix
6	Macintosh

the project advisor and a MQP Review team reviewer. The first result line is based on advisor responses and the second line is for reviewer responses.

Comment: The overlapping, but independent evaluations, of each project by advisor and reviewer allows a measure of validation of the MQP review process. The results show general agreement between advisor and reviewer evaluations except for the Computer Science and Math Level Demonstrated questions. The advisors were more likely to rate the computer science level at lower-level than the reviewers. Based on written advisor comments, the disparity in answers for the math level appears for two reasons. First, advisors are more familiar with the project and are aware of project aspects that may not be in the project report. Second, some advisors noted that the project involved "some" math, which may not have corresponded directly with the provided answers.

The professional ethics question could not be adequately evaluated by the reviewers so only advisor evaluations are shown. Only 4% of the projects demonstrated less than adequate ethics. Most projects required at least a moderate amount of programming effort and almost all students learned at least some amount of new tools and techniques. 85% of the projects showed at least the right amount of student effort level. This number is a bit improved from 81% in 1999.

2.14 Reviewer Project Evaluations

Additional numerical evaluations of the projects are shown in Table 12 based on the questions from the form in Appendix B. This form was completed by only a MQP Review team reviewer.

Comment: Problems with abstracts have dropped again from 10% in 1999 and 34% in 1997.

The report quality improved a bit with 13% of project reports less than adequate in 2001 compared to 19% in 1999. The number of reports judged to be excellent also improved.

The overall project quality is also better than 1999 when 16% were judged less than adequate while 8% were judged less than adequate in 2001. The number of excellent projects also increased.

Table 11: Advisor/Reviewer Evaluations by Percentage

		1	**2**	**3**	**4**	**5**	**avg.**
CS Level		1000	2000	3000	4000	grad	
Demonstrated	Adv.	0	0	21	64	13	3.8
	Rev.	0	0	11	87	2	3.9
Math Level		none	lin alg	prob/stat	4000	grad	
Demonstrated		47	11	30	9	4	2.1
		72	13	13	2	0	1.4
Demonstrated		poor		adequate		excellent	
Professional Ethics		0	4	28	45	19	3.7
				N/A			
Software Project Size/		none		moderate		large-scale	
Programming Effort		2	11	51	21	15	3.4
		4	11	38	38	9	3.4
Learned New		none		some		considerable	
Tools/Techniques/Info		0	9	11	45	36	4.1
		0	0	23	53	23	4.0
Project Objective met		unknown	no	mostly	yes	exceeded	
		0	9	32	40	17	3.6
		0	13	43	45	0	3.3
Overall Effort Level		too little		about right		too much	
(worth one unit/student)		2	13	49	32	2	3.1
		0	15	60	26	0	3.1

Table 12: Reviewer Project Evaluations by Percentage

	1	**2**	**3**	**4**	**5**	**avg.**
Abstract accurate and complete	missing	poor	adequate		excellent	
	2	2	55	38	2	3.4
Clearly stated project objective	poor		adequate		excellent	
	0	9	36	43	13	3.6
Quality of Background/	N/A	poor	adequate		excellent	
Literature Review	0	13	38	28	21	3.6
Style, grammar, spelling	poor		adequate		excellent	
	0	4	38	51	6	3.6
Project Methodology	unknown	poor	adequate		excellent	
Issues/Problems Discussed	0	13	36	38	13	3.5
Quality of report	poor		adequate		excellent	
	2	11	30	40	17	3.6
Quality of project	poor		adequate		excellent	
	2	6	43	30	19	3.6

2.15 Project as a Learning Experience

This question overlaps other questions on the review form about what students learned, their effort level, project quality and project strengths. Although on the review form, this question was not used and could be removed in subsequent reviews.

2.16 Project Strengths

Table 13 contains specific advisor and reviewer comments extracted from the evaluation forms concerning project strengths.

Comment: As in previous reviews, the projects were good when they were well-motivated, had a clear presentation indicating what was done, had a good design, and followed through on a particular topic.

Table 13: Project Strengths

Integrated a lot of software.
Outstanding research project.
Interdisciplinary.
Student maturation during the project.
Difficult topic.
Students exhibited great problem solving skills.
Met objectives of off-campus sponsor.
Self-motivated group.
Professional work.
Real development project.

2.17 Project Weaknesses

Table 14 contains specific advisor and reviewer comments extracted from the evaluation forms concerning project weaknesses.

Comment: As in previous reviews, projects with problems showed simplistic objectives, poor planning, and poor presentation of what was done. The most common problem were issues with the evaluation and testing portion of the projects.

Table 14: Project Weaknesses

Students had a hard time grasping experiments.
Weak evaluation.
Report gives no sense of task difficulty.
Students could have done more.
Poor project planning.
Students needed to make their contributions clearer.
Group did not function effectively as a team.
Lack of adequate effort by team.
Lack of user evaluation.
Project not well-focused.

2.18 Interdisciplinary Work

There were six projects involving other departments disciplines with six students from other departments as part the project teams. The disciplines represented by these projects were Biology (2), Biomedical Engineering and Electrical and Computer Engineering (2). One project was done in conjunction with the WPI Public Relations Office.

Comment: There were fewer interdisciplinary projects than in 1999, but in 2001 more students from other departments were involved in the projects.

2.19 Project Presentations

Oral presentations of the projects were done on MQP Presentation Day in April, 2001. A smaller group of projects were presented in December, 2000. During the presentations, faculty evaluated the content and presentation of each project through a evaluation process distinct from this MQP Review process. For completeness, results from that evaluation process are shown in Table 15, which contains a summary of the results on a scale from Strongly Disagree (SD) to Strongly Agree (SA) for each question. The results encompass more than 200 faculty evaluations.

Based on the presentations faculty generally evaluated both the content and presentation portions of each project to be good. Faculty evaluations show that "analysis and evaluation of results" was the most problematic aspect of the projects. Less than 10% of projects were evaluated to have any

Table 15: Faculty Evaluations of Project Presentations by Percentage

	NA	1 SD	2 D	3 Neu.	4 A	5 SA	ave.
Part I: Content							
Objectives clear	1	0	5	16	62	17	3.9
Motivation and impact discussed	0	0	7	21	57	15	3.8
Approach and methodology clear	0	1	3	27	54	14	3.8
Results analyzed and evaluated	2	4	17	26	38	12	3.4
CS material at 4000-level or higher	2	0	4	12	57	25	4.1
Independent learning demonstrated	6	0	1	15	55	23	4.1
Part II: Presentation							
Well organized	0	0	3	15	65	17	4.0
Professional speaking manner	0	0	7	27	50	16	3.7
High-quality visual aids	0	3	6	19	54	20	3.8
Understood & answered questions	3	1	3	19	60	13	3.8

Table 16: Expected Correlation Between Project Quality and Grade

	Project Quality				
Grade	1	2	3	4	5
C	X	X			
B		X	X	X	
A				X	X

problems for the presentation questions.

3 Analysis of Results

This section correlates various aspects of the MQPs with the evaluations the projects received. This analysis is intended to help identify which project characteristics tend to yield good projects and which traits result in lower quality projects.

3.1 Correlation of Evaluations

The following correlations show the relationship between various results and the project evaluations. The project grades and project evaluations are shown for all projects. Note: For sake of comparison the value 4 is assigned to an A project grade, a value 3 to a B project grade and a value 2 is assigned to a C project grade. Recall the project evaluations had a 1 to 5 range where 1 is poor, 2 is fair, 3 is adequate, 4 is good, and 5 is an excellent project. Because of the difference in these scales, the 1997 review team set the standard for correlation as shown in Table 16, suggesting that an A should never be rated less than a 4, a B should receive an evaluation of 2, 3, or 4, and a C should receive a 1 or a 2. Each entry with an "X" shows good correlation.

To start our analysis, we compare the two evaluation criteria taken from the reviewer questionnaire: project grade assigned by the advisor and the project quality (PQ). Table 17 shows the correlation between the project evaluation and the project grade assigned by the advisor. The projects were evaluated before obtaining the project grade.

Table 17: Correlation of Project Grade with Quality of Project

	Project Quality					
Grade	1	2	3	4	5	Total
C	0	2	0	0	0	2
B	**2**	2	21	4	**2**	32
A	0	**2**	**21**	26	17	66
Total	2	6	43	30	19	100

Comment: There is a disparity between the two evaluation measures for the projects. There are three cases to consider as again defined by the 1997 review team:

C1 The adviser and reviewer agree in their assessment of the project.

C2 The adviser graded too harshly or the reviewers overrated the project.

C3 The advisor graded too easily or the reviewer underrated the project.

The results show that while 73% (versus 72% in 1999) of the projects have correlating evaluations (C1), 25% (versus 25% in 1999) fall into case C3, and 2% (versus 3% in 1999) fall into case C2. In Table 17, cases C2 and C3 are represented by **bold** faced entries. These numbers indicate a similar correlation between reviewer rating and project grade as occurred in 1999.

For case C3, the reviewers agree that the quality of the projects is not entirely correlated with the individual grades assigned by the project advisor. One project (2%) received an A grade although it was assessed to be less than adequate. 21% of the A projects were rated as being adequate, but the A grade should be reserved for those projects that are more than adequate. Either the reviewers did not fully comprehend the significance of the work or the students and advisors agreed upon a less than adequate project. There is continued room for improvement here, and as the number of MQPs in our department grows, the faculty needs to pay attention to standardizing the quality and effort of all MQPs.

For case C2, one project that was rated as excellent received a B grade. This situation most likely reflects the reviewers judging the project better based on its report without being familiar with the process and detailed outcome of the project.

3.2 Correlation of Faculty Team Size and Evaluation

Table 18 shows the correlation between the number of faculty and the project evaluations. The two indicators are report quality (RQ) and project quality (PQ).

Table 18: Correlation of Faculty Team Size and Evaluation

Faculty Team Size	% of Projects	avg Grade	avg RQ	avg PQ
1	72	3.7	3.5	3.5
2+	28	3.5	3.8	3.8

Comment: The data show that single-faculty projects received better grades, but worse evaluations than multi-faculty projects. In 1999 single-faculty projects received both lower grades and ratings.

3.3 Correlation of Student Team Size and Evaluation

Table 19 shows the correlation between the number of students and the project evaluations.

Table 19: Correlation of Student Team Size and Evaluation

Student Team Size	% of Projects	avg Grade	avg RQ	avg PQ
1	28	3.5	3.2	3.2
2	21	3.7	3.9	3.9
3	40	3.6	3.7	3.7
4	11	3.8	3.6	3.4

Comment: As generally found in past MQP reviews, single-person projects resulted in the lowest average grade and evaluation.

3.4 Correlation of On/Off-Campus Projects and Evaluation

Table 20 shows the correlation between projects that were sponsored on/off-campus and the project evaluations.

Table 20: Correlation of On/Off-Campus Projects and Evaluation

Type	% of Projects	avg Grade	avg RQ	avg PQ
On	77	3.6	3.6	3.6
Off	23	3.9	3.5	3.5

Comment: There was little difference in the evaluated quality between projects that were either off-campus or associated with an organization when compared with on-campus projects. However, off-campus projects received higher average grades. One explanation is that off-campus projects may have constraints imposed by the sponsor. Working within these constraints may be an accomplishment for the project that is hard to account for in the evaluation.

3.5 Correlation of Project Duration and Evaluation

Table 21 shows the correlation between the registered units for a project and the project evaluations.

Table 21: Correlation of Registered Units and Evaluation

Registered Project Units	% of Projects	avg Grade	avg RQ	avg PQ
1	51	3.7	3.6	3.5
1 1/6+	36	3.5	3.6	3.7
unknown	13	3.7	3.5	3.7

Comment: In the past, projects that were completed with more than one unit of work typically evaluated lower. While the grades show evidence of this trend, the project evaluations do not. The results do show that only about 50% of the projects are known to have been completed with one unit of registered work.

Table 22 shows the correlation between the project duration measured in terms and the project evaluations.

Table 22: Correlation of Project Duration (Terms) and Evaluation

Project Duration in Terms	% of Projects	avg Grade	avg RQ	avg PQ
1	6	4.0	4.0	3.7
2	17	3.8	3.8	3.8
3	32	3.5	3.4	3.3
4+	32	3.6	3.7	3.7
unknown	13	3.7	3.5	3.7

Comment: There appear to be no particular correlation between number of registered terms and the project grade/evaluation.

3.6 Correlation of Project Report Size and Evaluation

Table 23 shows the correlation between the project report size and the project evaluations. The report size does not include code and appendices.

Table 23: Correlation of Project Report Size and Evaluation

Project Report Size	% of Projects	avg Grade	avg RQ	avg PQ
0–39 pgs.	26	3.5	3.1	2.8
40–69 pgs.	45	3.6	3.3	3.4
70+ pgs.	30	3.9	4.4	4.5

Comment: The results of this correlation show that evaluations and grades correlate to the size of the project report. This result has generally been the case in previous reviews as shorter reports indicate that students did not accomplish much or that they did not allocate enough time to write an adequate report.

3.7 Correlation of Computer Science Level and Evaluation

Table 24 shows the correlation between the Computer Science level and the project evaluations. The level provided by project advisors is used for this correlation.

Table 24: Correlation of Computer Science Level and Evaluation

Computer Science Level	% of Projects	avg Grade	avg RQ	avg PQ
3000	21	3.0	2.9	2.9
4000	64	3.8	3.6	3.6
grad	13	3.8	4.7	4.8

Comment: The project grade and evaluations are highly correlated to the CS level, as would be expected.

3.8 Correlation of Math Level and Evaluation

Table 25 shows the correlation between the math level and the project evaluations again using the evaluations from project advisors.

Table 25: Correlation of Math Level and Evaluation

Math Level	% of Projects	avg Grade	avg RQ	avg PQ
none	47	3.5	3.3	3.2
lin alg	11	4.0	4.0	3.6
prob/stat	30	3.8	3.9	4.1
4000	9	3.2	3.2	3.5
Grad	4	4.0	4.0	4.0

Comment: Projects that involved some math generally received better grades and evaluations. Part of the reason may be that stronger students are taking on these projects. Another consideration is that the topic requires more effort.

3.9 Correlation of Overall Effort Level and Evaluation

Table 26 shows the correlation between the overall effort level and the project evaluations again using evaluations of the project advisors.

Table 26: Correlation of Overall Effort Level and Evaluation

Overall Effort Level	% of Projects	avg Grade	avg RQ	avg PQ
1 too little	2	2.0	2.0	2.0
2	13	3.2	3.0	2.8
3 about right	49	3.6	3.4	3.4
4	32	4.0	4.2	4.1
5 too much	2	4.0	4.0	5.0

Comment: As expected, there is a strong correlation.

4 Assessment of Student Learning Outcomes

At the time this report is being written, the Computer Science Department has a draft for the Computer Science Undergraduate Program objectives and outcomes. Although not yet completed nor approved, relevant draft outcomes are listed below with an assessment of the degree of success in demonstrating each of these outcomes using the results of the MQP review process. The MQP is only one avenue for students to demonstrate desired program outcomes so it is not expected that all MQPs need to demonstrate all outcomes. However, the MQP is the capstone experience in our curriculum and should demonstrate many of the program outcomes for many of the students.

1. **Almost all students will demonstrate an understanding of Software Engineering principles and the ability to apply them**

to software design.

In terms of software design and implementation of a specific solution, 87% of the projects (88% of students) demonstrated this type of work. Software engineering was explicitly listed as a relevant area for 34% of the projects (34% of students).

2. **All students will have demonstrated upper-level knowledge of Computer Science topics.**

   Advisors rated 79% of projects (83% of students) as exhibiting 4000-level work or higher.

3. **All students will demonstrate an ability to analyze the behavior of computational systems.**

   43% of the projects (45% of students) were identified as containing analysis work. 40% of the projects (45% of students) were identified as containing performance evaluation work. 50% of faculty evaluations of project presentations agreed that the project included analysis and evaluation of results.

4. **All students will demonstrate a knowledge of probability or statistics.**

   Advisors rated 30% of projects (30% of students) as containing math at the level of probability/statistics. They rated an additional 13% of projects (12% of students) as containing a higher level of math.

5. **All students will demonstrate independent learning.**

   45% of the projects (42% of students) were identified as containing research work. Students learned new tools and techniques for 100% of the projects. For 81% of the projects (81% of students) learned more than "some" amount.

6. **All students will demonstrate the ability to locate and use technical information from multiple sources.**

   Students learned new tools and techniques for 100% of the projects. For 81% of the projects (81% of students) learned more than "some" amount.

7. **All students will demonstrate the ability to design experiments and interpret experimental data.**

   40% of the projects (45% of students) were identified as containing performance evaluation work. 32% of the projects (46% of students) were identified as containing data collection work.

8. **All students will demonstrate an understanding of professional ethics.**

   Advisors rated 96% of projects (93% of students) as demonstrating adequate or better professional ethics.

9. **All students will participate in a class or project team.**

   91 of the 104 computer science project students (88%) in the MQP review participated as part of a team in completing the project.

10. **Almost all students will have demonstrated the ability to communicate effectively in speech.**

    In less than 10% of cases, did faculty find any problems with project presentations.

11. **All students will have demonstrated the ability to communicate effectively in writing.**

    The reports for 92% of projects (88% of students) were evaluated to be adequate or better.

5 Conclusions and Recommendations

The 2001 review of Computer Science MQPs reflects data and evaluations for 47 MQPs, involving 104 computer science students, that were completed between the Summer of 2000 and the Spring of 2001. In this section, we attempt to draw some conclusions from the data collected during the evaluation process.

5.1 Quality of Project

The overall project quality shows that many more projects were judged as at least adequate (92%) in this year's review compared to the previous year with a similar correlation between project evaluations given by the reviewers and project grades assigned by the advisor.

Most of the MQPs were good capstone learning experiences for CS majors and meet the educational goals of the department. There was some concern on a few of the projects as good learning experiences. These problematic projects showed little rationale for choosing the design, displayed a lack of consideration for alternatives or indicated the students did not expend enough effort.

85% of the MQPs were judged to involve at least an adequate amount of student effort. Typical Computer Science MQPs include the design and implementation of a large piece of software with many following the software life cycle from requirements gathering to implementation. Unfortunately not enough had results on testing and evaluation of the work.

5.2 Quality of Report

The quality of the reports themselves was better than in the previous review. Fewer reports were judged to be less than adequate with more reports also being judged as excellent. Unlike previous reviews, the report quality for projects done in one or two terms was not worse than projects done during a longer period of time.

5.3 Students per MQP

The number of single student CS MQPs returned to 28% (versus 35% in 1999, 28% in 1997, 44% in 1995 and 42% in 1993). The average number of students per project increased again to 2.3 versus 2.0 in 1997 and 2.1 in 1999. This trend is important and shows that the faculty are successfully able to group students together on projects as the number of students needing projects has grown. The results show that single-student projects received the lowest grades and evaluations.

5.4 Distribution of CS Faculty over MQPs

There is some improvement in both measures of faculty loading in regards to number of projects advised (Table 2) with the Gini Coefficient dropping slightly from 0.406 in 1999 to 0.402 in 2001. The results in Table 3 show a decrease from 0.453 in 1999 to 0.399 in 2001. We think a value of 0.2-0.28 is a more appropriate measure, although we also need to consider the impact of project centers, which often result in all of the projects advised by the same faculty member.

5.5 Project Resources

The project data show that Java, C, C++ and Perl were the most used programming languages for projects. PCs were used for the majority of projects with most of these running the Windows platform, but with a number using the Linux platform.

The evaluation process generally worked reasonably well. Extending the process to included faculty advisors was a success as all faculty returned their project evaluation forms and these forms allowed more accurate information about the projects to be included in this report. The development of a review process for MQP presentations also provided better assessment for this aspect of the MQP experience. A slight improvement in future reviews would be to correlate presentation data on a per-project basis rather than use it in aggregate form.

While the review forms worked well and provided a substantial amount of data, some revisions are needed for future reviews, particularly in light of student learning outcome evaluation. These include:

- The question concerning "Math Level Demonstrated" needs to include a response for "some" math, which was not reflected in the current set of possible responses.

- The question about the types of work encompassed by the project needs work. For example, there was not an explicit response for "testing." There was confusion with responses for both "Design" and "Design/Implementation." The types of project activity also needs to be better aligned with the set of student learning outcomes.

- There was not a question that correlated well with the independent learning outcome.
- The question about "the project as a good learning experience" can be dropped as it overlaps with other review questions.
- The department needs to consider involvement of external (to the department) professionals, such as alumni, in the review process.

5.6 Recommendations for Improving CS MQPs

The following list of recommendations are drawn from the analysis and conclusions of this Computer Science MQP Peer Review.

- Increase student team size and avoid single-student projects when possible. Increasing enrollments in the department make this recommendation a necessity. Better mechanisms for bringing project groups together earlier need to be investigated. Working in project groups improves cooperative and communication skills of the students. Larger MQP teams offer more efficient use of a faculty member's time.
- Emphasize the testing and evaluation phase. Lack of adequate evaluation by external sources was a problem with many of the design and implementation projects. Serious analysis of projects is a general weakness in department MQPs. More formal analysis would also increase the level of mathematics and statistics displayed by the projects.
- Emphasize the need for students to indicate why the MQP was a good experience, what was difficult about the project and what experiences/courses the MQP builds upon. It was difficult with some projects for the reviewers to understand the significance of the work and upon which prior student work the project built upon.
- Strive to have MQPs build on previous MQPs and projects. In industry, our graduates will have to learn how to work with old code from old projects, and one way we can address this is through building upon previous MQPs and theses. This approach makes faculty more efficient and creates a pipeline of projects so the students can see the larger objective for their individual project.

References

[1] George T. Heineman and Robert E. Kinicki. 1997 computer science department MQP review. Technical Report WPI-CS-TR-97-07, Worcester Polytechnic Institute, August 1997.

[2] Micha Hofri and Craig E. Wills. 1999 computer science department MQP review. Technical Report WPI-CS-TR-99-24, Worcester Polytechnic Institute, August 1999.

[3] Robert E. Kinicki and Craig E. Wills. Computer science department MQP review. Technical Report WPI-CS-TR-91-13, Worcester Polytechnic Institute, July 1991.

[4] Robert E. Kinicki and Craig E. Wills. 1993 computer science department MQP review. Technical Report WPI-CS-TR-93-5, Worcester Polytechnic Institute, August 1993.

[5] Robert E. Kinicki and Craig E. Wills. 1995 computer science department MQP review. Technical Report WPI-CS-TR-95-1, Worcester Polytechnic Institute, August 1995.

A Advisor and Reviewer Form

The following form was used by advisors to evaluate all MQP projects. This form was also used by the reviewers.

CS Level Demonstrated	1 1000	2 2000	3 3000	4 4000	5 grad
Math Level Demonstrated	1 none	2 lin alg	3 prob/stat	4 4000	5 grad
Demonstrated Professional Ethics	1 poor	2	3 adequate	4	5 excellent
Software Project Size/ Programming Effort	1 none	2	3 moderate	4	5 large-scale
Learned New Tools/Techniques/Info	1 none	2	3 some	4	5 considerable
Project Objective met	1 unknown	2 no	3 mostly	4 yes	5 exceeded
Overall Effort Level (worth one unit/student)	1 too little	2	3 about right	4	5 too much

1. Circle the following types of work and areas of computer science that are relevant for this project.

   Analysis
   Data Collection (Empirical)
   Design
   Design/Implementation
   Performance Evaluation
   Research
   Simulation
   Survey
   Other________________________

   AI
   Theory/Foundations
   Architecture
   Networks
   DataBase
   Webware
   Graphics
   HCI
   Languages/Compilers
   Software Engineering
   Operating Systems
   Distributed Systems
   Other________________________

2. Circle the following software languages, tools, and hardware resources used for this project.

C
C++
Assembly Lang.
Lisp/Scheme
Java
Perl/Tcl/Tk
Other____________________

Macintosh
PC/Windows
PC/Linux
CS Unix
CCC Unix
Other____________________

3. Project strengths/weaknesses/other comments?

B Reviewer-Only Form

The following form was used only by the reviewers to evaluate all MQP reports.

1. Number and department of MQP student(s)__________________
2. Final grade given to report
3. Terms to complete MQP__________________ Units Earned__________________
4. Report length in pages (excluding appendices and code)__________________

Abstract accurate and complete	1 missing	2 poor	3 adequate	4	5 excellent
Clearly stated project objective	1 poor	2	3 adequate	4	5 excellent
Quality of Background/ Literature Review	______ no. refs	2 poor	3 adequate	4	5 excellent
Style, grammar, spelling	1 poor	2	3 adequate	4	5 excellent
Project Methodology Issues/Problems Discussed	1 unknown	2 poor	3 adequate	4	5 excellent
Quality of report	1 poor	2	3 adequate	4	5 excellent
Quality of project	1 poor	2	3 adequate	4	5 excellent

1. Was this project a good learning experience? What was learned by the student(s)?
2. Project strengths: Project weaknesses:
3. Was this project a continuation of an earlier project, and if so, did the students indicate the part of the work that is theirs?
4. Did this project involve any interdisciplinary work? What departments or organizations were involved? Off-campus or on.